REMARKS

Claims 1-8 and 10-24 were pending in the present application. Claims 5-8, 14, 16-18, 22, and 23 have been cancelled. Claims 1, 10, 20, 21, and 24 have been amended. Accordingly, claims 1-4, 10-13, 15, 19-21, and 24 are now pending in the application.

Claims 20-24 stand rejected under 35 U.S.C. 101 for being directed to non-statutory subject matter. Applicant respectfully submits "carrier medium" is statutory subject matter in the same way that "computer readable medium" is statutory. However, to expedite allowance, Applicant has amended claims 20, 21, and 24 to be directed to a computer readable medium as requested by the Examiner.

Claims 1-8 and 10-24 stand rejected under 35 U.S.C. 103 as being unpatentable over Arimilli et al. (U.S. Patent Number 6,480,975, hereinafter "Arimilli '975") in view of Arimilli et al. (U.S. Patent Number 5,867,511, hereinafter "Arimilli '511"). Applicant respectfully traverses this rejection.

Applicant's claim 1 has been amended to include features that were previously recited in dependent claims 5-8. Accordingly, claim 1 recites

> "A method, comprising:
> detecting an error in data stored in a directory cache in a system;
> determining if the detected error is correctable;
> while the system is in operation:
> making at least a portion of the directory cache unavailable to one or more resources in the system in response to determining that the error is uncorrectable, wherein making at least the portion of the directory cache unavailable comprises generating a cache miss in response to a request to access the directory cache;

testing the at least a portion of the directory cache while the at least a portion of the directory cache is offline based on determining that the error is uncorrectable;

servicing the at least a portion of the directory cache in response to testing the directory cache; and

dynamically placing the allowing access to the at least a portion of the directory cache in response to servicing the at least a portion of the directory cache."

The Examiner acknowledges Aramilli '975 does not teach generating a cache miss in response to a request to access the directory cache. However, the Examiner asserts Arimilli '511 teaches this feature. The Examiner further asserts the combination of Arimilli 975 and Arimilli '511 teaches the limitations recited in Applicant's claims 5-8. Applicant respectfully disagrees with the Examiner's characterization of Arimilli '975 and Arimilli '511.

Specifically, Arimilli '511 teaches at col. 7, line 11 through col. 8, line 6

> "It can be seen that repair mask 76 is a convenient means for both keeping a defective cache line from **ever** indicating a cache hit and keeping a defective cache line from **ever** being chosen as a victim. Repair mask 76 can accordingly be used in place of the bit line redundancy and word line redundancy provided in prior art cache components. ... Also, by using repair mask 76, all available cache lines are used, instead of some (redundant) cache lines never used, making more overall efficient use of the cache. Repair mask 76 further provides these advantages without any re-routing overhead, and without requiring "fuse-blow" for the directory array, LRU array, or cache array. This, in turn, allows significantly faster cache operation and significantly reduced manufacturing cost.
>
> This novel method of using functional masking to bypass defects in caches eliminates the performance degradation and the silicon area increase of the standard cache defect repair method. From a functionality perspective, certain congruence classes may be effectively running 6-way or 7-way set associative (instead of the intended 8-way set associative). However, due to the statistical nature of cache behavior, this reduction in associativity for certain congruence classes is typically unnoticeable at the user level.
>
> The use of a repair mask additionally allows for dynamic cache defect

bypassing (of locations in the caches that are generating errors) by updating the repair mask real-time when the errors are detected. The cache lines may be tested initially at fabrication and any noted defects can be handled by permanently setting the value of the corresponding field in the repair mask. Thereafter, each time the computer is booted (turned on), the mask might be automatically updated based on firmware testing, as part of the boot process. Finally, the repair mask can be updated upon detection of directory parity errors, cache entry array ECC errors, or LRU errors. A hardware algorithm could be provided to set the values in the repair mask array. For example, one 2-bit field could be provided in the repair mask for each cache line. The 2-bit field may initially be set to zero, and incremented each time a error is detected on that cache line. This allows the 2-bit field to act as a counter, setting the cache line as defective only when three cumulative parity errors have been recorded for a given cache line.

In order to continue to reliably run the processor after encountering defective cache locations, when the repair mask entry associated with the line in the cache is set to indicate the line is defective, the contents of the cache at that location are flushed. Once the repair mask entry has been set, **any future accesses** to that cache line will be forced by the repair mask to see a miss on that line, and the line would **never** be re-used (victimized). This solution has practically no overhead when compared to prior art schemes, such as redundant lines. It is also particularly useful in those applications where the processors operate in harsh environments but must continue to function in the event of run-time defects." (Emphasis added)

In addition, Arimilli '975 teaches at col. 6, lines 2 – 14, "This output is connected to a retry circuit 82 and an ECC circuit 84. Retry circuit 82 causes the cache operation to be repeated after a delay sufficient to allow ECC circuit 84 to complete its operation. ECC circuit 84 uses all of the bits from all address tags in congruence class 74, and further uses bits from a special ECC field 86. Only one ECC field is provided for each congruence class, rather than providing one ECC field for each cache block. When ECC circuit 84 operates on the input values, it generates corrected values which are fed back to the cache blocks and ECC field of the congruence class. If a double-bit error is detected, operation of the processing unit can be halted using an appropriate circuit 88." (Emphasis added)

From the foregoing, Arimilli '511 teaches being able to overcome cache defects within a processor by never again using the defective cache line. More particularly, Arimilli '511 teaches testing the cache during manufacturing and at boot-up. Further,

Arimilli '511 teaches permanently causing a cache miss in response to an access request to a bad cache line. Thus, Applicant submits Arimilli '511 teaches away from Applicant's claimed invention, since Applicant's invention is directed toward taking the directory cache offline for testing during continued operation of the domain in response to an error, requesting service of the directory cache, and returning the directory cache to an online status in response to the service. Arimilli '975 teaches halting the processor in response to an uncorrectable error, which would be unacceptable in Applicant's system.

Thus, neither Arimilli '511 nor Arimilli '975, teach or suggest "while the system is in operation: testing the at least a portion of the directory cache while the at least a portion of the directory cache is offline based on determining that the error is uncorrectable," nor "servicing the at least a portion of the directory cache in response to testing the directory cache," nor "dynamically placing the allowing access to the at least a portion of the directory cache in response to servicing the at least a portion of the directory cache" as recited in Applicant's claim 1.

Thus, neither Arimilli '511 nor Arimilli '975, taken either singly or in combination, teach or suggest the combination of features recited in Applicant's claim 1. Accordingly, Applicant submits claim 1, along with its dependent claims, is believed to patentably distinguish over Arimilli '975 in view of Arimilli '511 for the reasons given above.

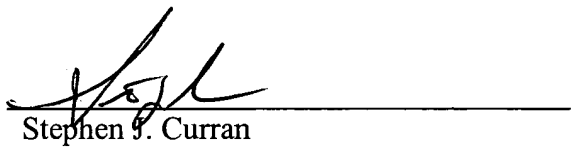
Applicant's independent claims 10 and 20 recite features that are similar to the limitations recited in claim 1. Thus, Applicant submits claims 10 and 20, along with their respective dependent claims, patentably distinguish over Arimilli '975 in view of Arimilli '511 for at least the reasons given above.

CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-55600/SJC.

Respectfully submitted,



Stephen J. Curran
Reg. No. 50,664
AGENT FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: April 11, 2005